

Elenco di esercizi C+Unix

Enrico Bini

October 14, 2020

Premessa

Segue un elenco di esercizi assegnati durante l'anno.

Importante Per massimizzare l'apprendimento, si raccomanda di leggere la soluzione soltanto **dopo** aver provato a risolvere l'esercizio autonomamente. Difatti, leggere una soluzione scritta da altri dopo aver provato a risolvere il problema da soli permette di capire meglio la soluzione proposta.

Esercizi C

Es. [print-after] Si scriva un programma che legge due stringhe di caratteri (**s1** e **s2**) di lunghezza massima di 80 caratteri mediante `fgets`.

Preliminarmente, il programma “chiude” entrambe le stringhe **s1** e **s2** scrivendo il byte 0 sul primo byte non stampabile (un byte è stampabile se ha codice ASCII compreso fra 32 e 126). (questo serve per eliminare un eventuale byte di “a capo”)

- Se **s2** è contenuta all'interno di **s1**, il programma stampa la parte di **s1** che segue **s2**.
- Se **s2** non è contenuta all'interno di **s1**, non stampa niente.

Per esempio, se le stringhe **s1** e **s2** sono rispettivamente:

```
Ciao a tutti  
ia
```

allora verrà stampato

```
o a tutti
```

Es. [count-char] Si scriva un programma che legga una stringa da `stdin` e, per ogni carattere presente nella stringa, scriva a `stdout` una riga con il numero di occorrenze del carattere nella stringa e il carattere stesso. Per esempio, se la stringa letta è

```
Ciao a tutti!!
```

venga stampato a `stdout`

```
2,  
2,a  
2,i  
1,o  
3,t  
1,C  
1,u  
2,!
```

in ordine a piacere.

Es. [get-exponent] Si scriva un programma che:

1. legga un `double` da tastiera,
2. estraiga l'esponente della sua rappresentazione in floating point secondo lo Standard IEEE 754-1985
3. stampi tale esponente in decimale.

Suggerimento: si provi a leggere la memoria dove il numero floating point è memorizzato, come un intero `unsigned long` da cui poi estrarre l'esponente attraverso la manipolazione dei suoi bit.

Es. [binary] Si scriva un programma che legge un intero senza segno da tastiera `stdin` e scrive sul terminale la sua rappresentazione in base 2. Si eviti di usare gli operatori di divisione `/` e di resto `%`, preferendo invece gli operatori bitwise e quelli di shift.

Es. [exam-2019.01.28] Implementare la funzione con prototipo

```
int range_of_even(int * nums, int length, int *min, int *max);
```

La funzione ha quattro parametri:

- `nums` è un array di numeri interi;
- `length` è la dimensione di `nums`;
- `min` e `max` sono puntatori usati dalla funzione per restituire degli interi al chiamante.

La funzione deve determinare il valore massimo e minimo **dei valori pari** presenti in `nums`. Se tali limiti esistono allora la funzione deve restituirli tramite i puntatori `min` e `max` al chiamante e restituire 1. Se l'array non contiene alcun numero pari, la funzione deve restituire 0 e i valori in `*min` e `*max` non saranno significativi.